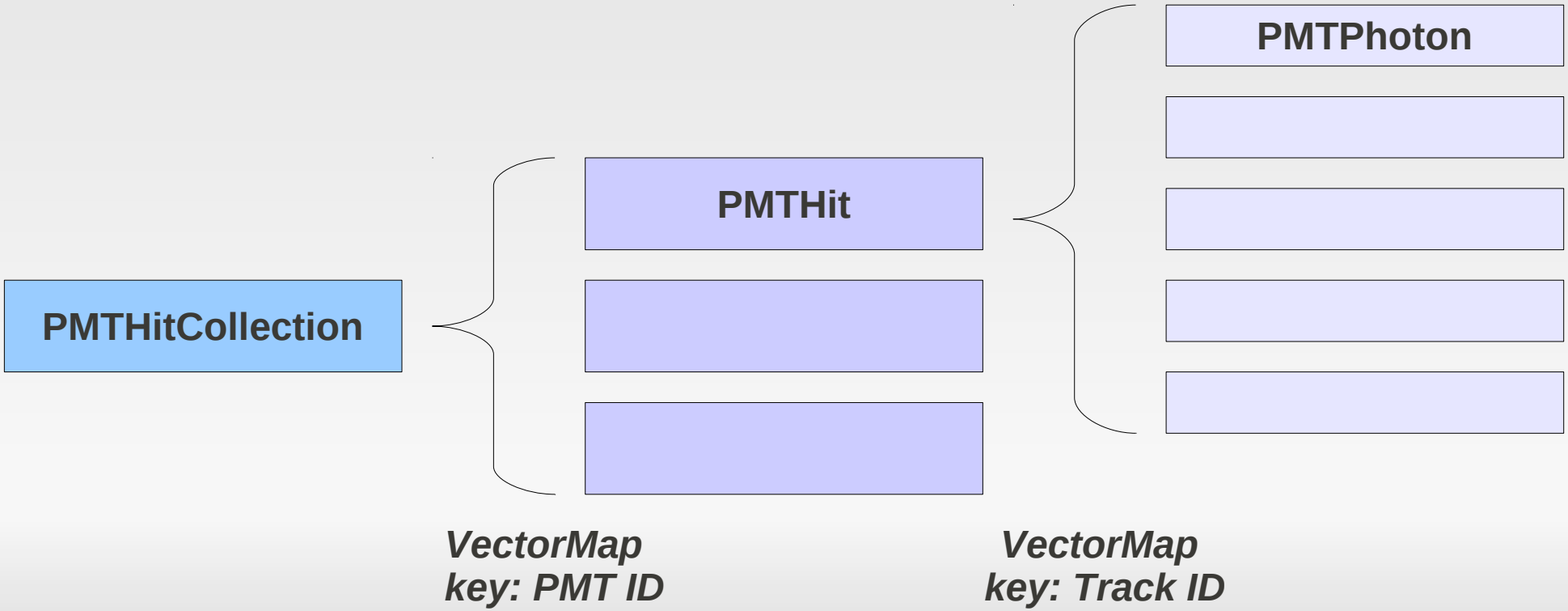
Optical MC Update
06 / 09 / 2010

Ben Jones, MIT

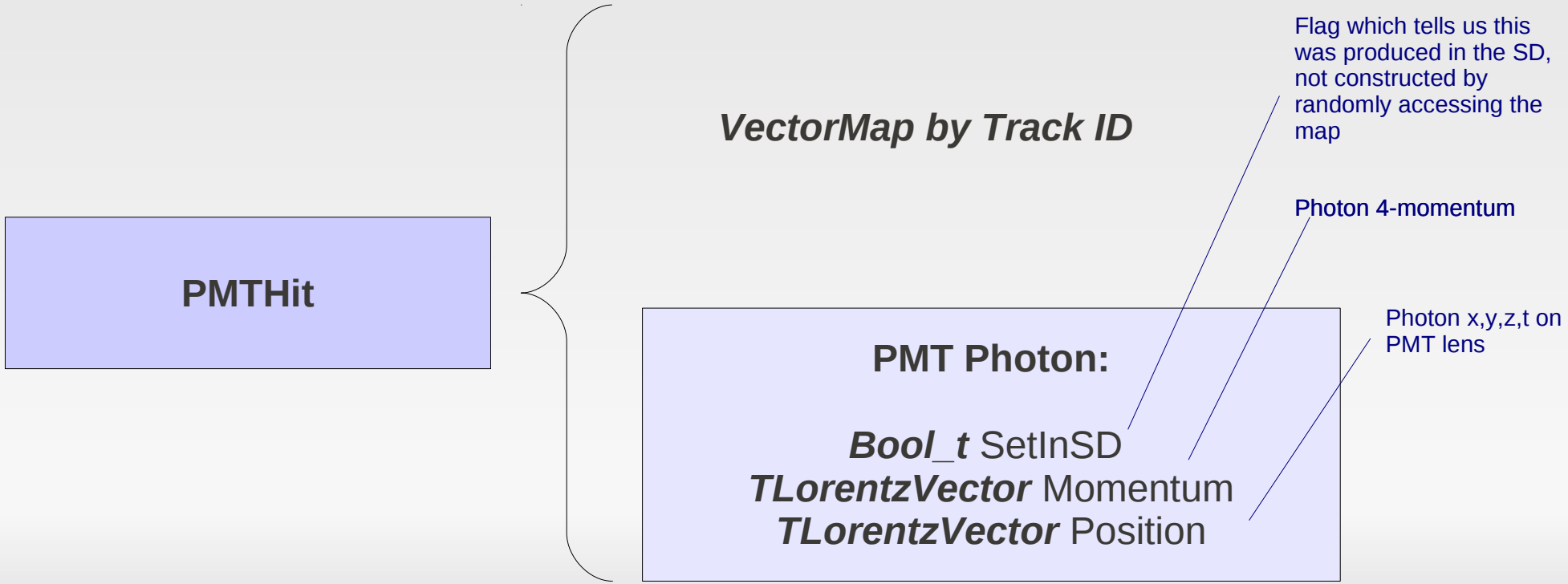
1) New PMTHit Structure

- Significantly changed the design since last time, taking on board Bills comments about not using the GEANT4 base classes
- Also simplified the class structure – no abstract base class, just simple data container. For any practical pulse reconstruction, same photon data is always required anyway.

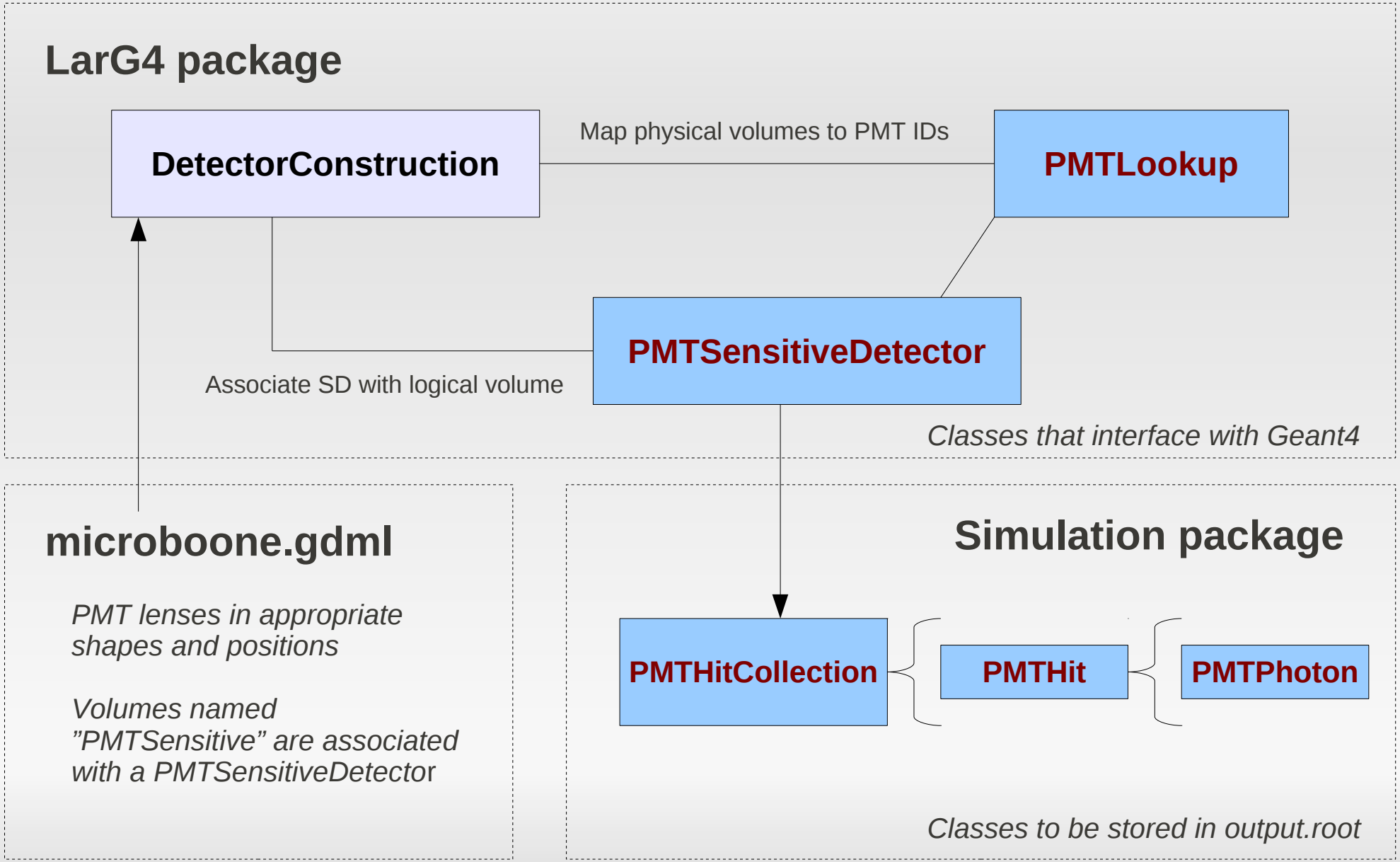


PMTHit Data Content

- **One PMTHit per PMT per event**
- Each PMTHit contains data about **all photons** which stepped into the sensitive volume
- Each PMTHit then used to generate PMT digi further along the simulation chain. This part has not yet been started.



PMT Framework



PMT Implementation Details

- Detector Construction Time:
  - Parse GDML as normal.
  - Loop through physical volume store looking for a volume named PMTSensitive. If one is found, associate a PMTSensitiveDetector with it.
  - If such a volume was found, loop through the physical volume store looking for placed copies of this volume. Every time one is found, assign it a unique name and a PMT ID, and add it to the singlet PMTLookup table.
- Start of Event
  - PMTSensitiveDetector produces a new PMTHitCollection for itself.
- During Event
  - If a photon steps into a volume with a PMTSensitiveDetector, kill the track and produce a PMTPhoton object. Find the name of the physical volume and look up the PMT ID in the PMTLookup table. Store the PMTPhoton in the relevant PMTHit of the PMTHitCollection.
- End of Event
  - LarG4 looks for PMTSensitiveDetectors in the Geant4 sensitive detector manager. If one is found, it is asked for its PMTHitCollection, which is stored in the event.

2) New PMT Geometry

Previous (prototype) PMT Geometry:

1) TPBPlate

Shape:
Cylinder

Material:
TPB

Physics:
OpWLS Process in OpticalPhysics constructor

2) PMTLens

Shape:
Radius 15cm
Depth 5cm

Material
LAr *

Physics:
PMTSensitiveDetector associated in DetectorConstruction

Updated Geometry

2) AcrylicPlate

Shape:
Cylinder

Material:
Acrylic

4) PMTUnderside

Shape:
Ellipsoid

Material:
Solid Glass

5) PMTStalk

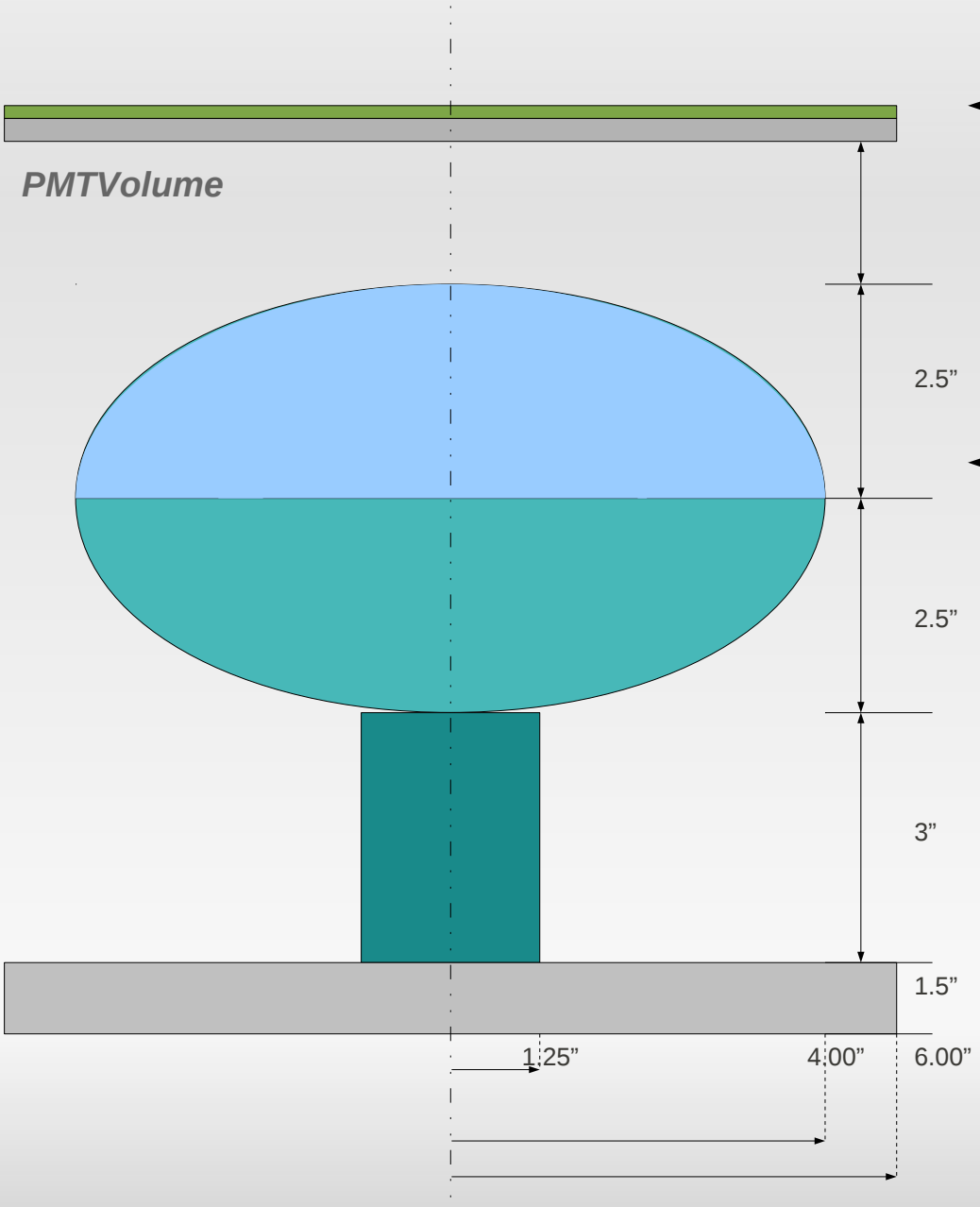
Shape:
Cylinder

Material:
Solid Glass

6) PMTSteelBase

Shape:
Cylinder

Material:
Stainless Steel



1) TPBCoating

Shape:
Cylinder

Material:
TPB

Physics:
OpWLS Process in OpticalPhysics constructor

3) PMTLens

Shape:
Ellipsoid

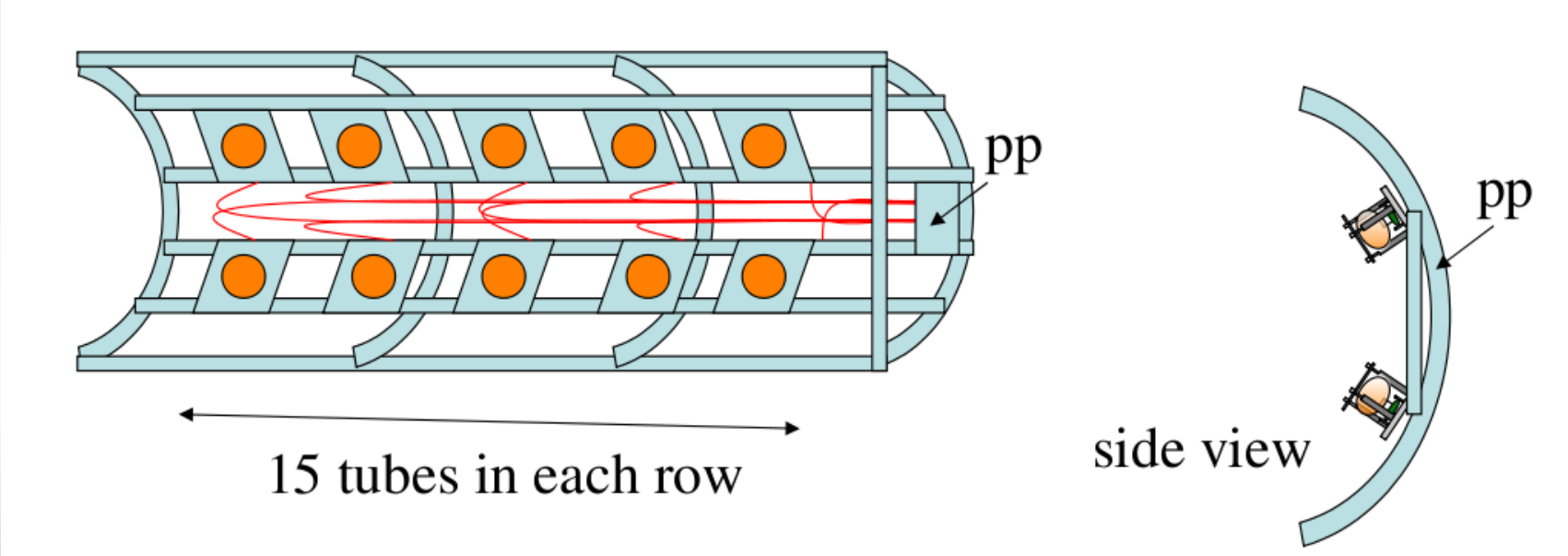
Material:
LAr *

Physics:
PMTSensitiveDetector associated in DetectorConstruction

* PMT Lens material defined as LAr so photon travels unimpeded into sensitive detector area. From here we can use photon position and momentum and measured PMT properties to decide if it is detected.

PMT Positions

- An undergrad in our group, Tess Smidt, has been working on determining PMT positions within the cryostat to fit between support beams, etc
- I will be adding PMT placement to the TPC definition scripts in the Geometry package rather than hand writing the gdml as I have been thus far
- The positions provided by Tess in the coming weeks will be a first attempt at reallistic PMT placement
- Right now, I have a single line of 10 PMTs along one side for code testing.



3) Progress Towards PhotonLibrary

- Recall : Simulating every photon per event takes far too long (hours per event)
- Aim is to develop a library of PMT responses for sctinillation photons produced at any point in the detector
- To build the library, need a controllable, isotropic sample of 1000s (10000s?) of photons per voxel.
  (Note – optical voxels not necessarily the same as drift voxels)
- This task requires a custom event generator.
  (closest existing event generator would be the single particle generator – but 1 photon per evt brings too much overhead)
- I have written **evgen::LightSource** for this purpose

LightSource Event Generator

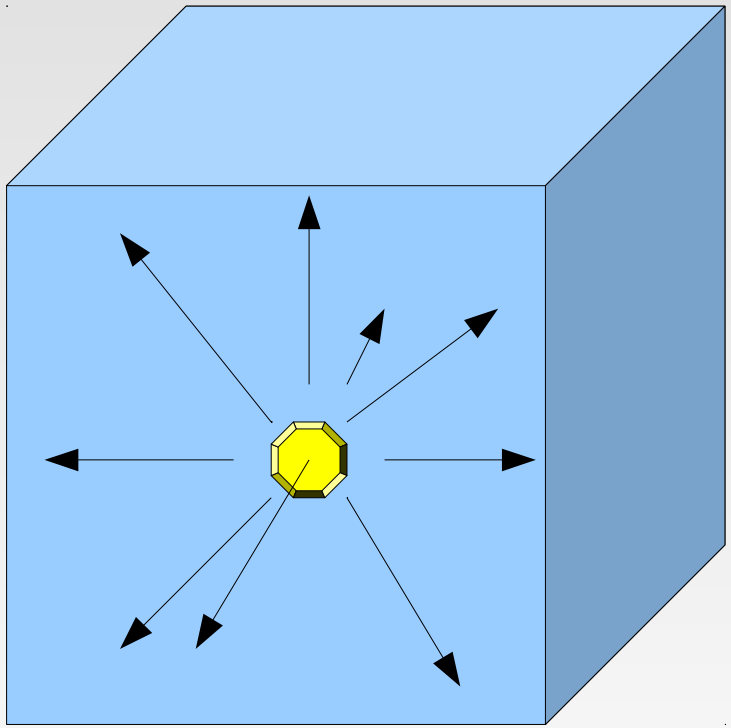
- Event generator which simulates an extended, isotropic light source at some position in the detector
- Two modes, as described on next two slides:

  - Scan Mode

  - File Mode

- In both modes, can optionally produce a ttree of
  - photon positions
  - photon momenta
  - event ID

  To store in the histos.root file



(Code loosely based on the single particle event generator)

LightSource in Scan Mode

- **Step through the detector in x,y,z**
- Choose:
  - voxel size
  - number photons per event
  - photon momenta and spread
  - production time and spread
  - distributions to sample p,t,x (uniform / gaussian)
- Read from geometry:
  - size and position of TPC

```
<config name="LightSource" version="scanmode" base="default">

      Scan mode : positions are chosen by stepping through the detector volume
                  systematically in x,y,z.  Times and momenta are randomly chosen
                  based on supplied parameters.  A specified number of protons
                  are shot isotropically from each spot.  This is source mode 1.

  <param name="SourceMode">  <int>    1                 </int>    </param>

      Generate a TTree of photon poisitons in histos.root?

  <param name="FillTree">    <bool>   true              </bool>  </param>

      The centre and width of the distribution of photon momenta for this
      light source.  Units are eV.

  <param name="P">           <float>  9.7               </float>  </param>
  <param name="SigmaP">      <float>  0.1               </float>  </param>


      The number of steps to be taken in x,y and z across the TPC volume

  <param name="XSteps">      <int>    100               </int>    </param>
  <param name="YSteps">      <int>    100               </int>    </param>
  <param name="ZSteps">      <int>    100               </int>    </param>

      The centre and width of the distribution of photon production times

  <param name="T0">          <float>  0.0               </float>    </param>
  <param name="SigmaT">      <float>  0.0               </float>    </param>

      The number of photons to shoot per location

  <param name="N">           <int>    10                </int>    </param>

      The following determine how random numbers should be chosen for position,
      momentum and time coordinates.  In each case, 0 throws a uniform distribution
      and 1 throws a gaussian.  For positions, the width is the step length.
      For T and P, the width is supplied in this config.

  <param name="PosDist">     <int>    0                 </int>    </param>
  <param name="PDist">       <int>        0                 </int>    </param>
  <param name="TDist">       <int>    0                 </int>    </param>

</config>
```

LightSource in FileMode

- **Place light souce at custom positions in the detector, specified by text file**
- Set in config:
  - File path
  - distributions to sample p,t,x (uniform / gaussian)
- Set in text file *(1 line / event)*
  - position, extent of light source
  - production time, spread
  - photon momenta, spread
  - number of photons
- Read from geometry
  - centre of TPC (origin)

```
<config name="LightSource" version="filemode" base="default">

      File mode : positions, times, number of particles are read from a
                  text file. This is source mode 0.

  <param name="SourceMode"> <int>    0                <int>    </param>

      Generate a TTree of photon poisitons in histos.root?

  <param name="FillTree">    <bool>   true             </bool>   </param>

      The name of the data file to read

  <param name="FileName">    <string> "LightSource.txt"  </string> </param>

      The following determine how random numbers should be chosen for position,
      momentum and time coordinates.  In each case, 0 throws a uniform distribution
      and 1 throws a gaussian, with centres and widths given by the parameters
      supplied either in the config (for P) or text file (for X,Y,Z,T).

  <param name="PosDist">     <int>    0                <int>    </param>
  <param name="PDist">       <int>       0                 </int>    </param>
  <param name="TDist">       <int>    0                <int>    </param>


</config>
```

	x	y	z	t	dx	dy	dz	dt	p	dp	n
2	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	9.7	0.1	1000
3	1.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	9.7	0.1	1000
4	2.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	9.7	0.1	1000
5	3.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	9.7	0.1	1000
6	4.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	9.7	0.1	1000

Next : LibraryBuilder, PMTDigi, PropagatePhotons

- LightSource event generator → PMT response Library

  Quite a simple analyzer can do this for us. But how to go from PMT Photons to PMT pulse shapes, and which should be stored in the library is still unclear.

- Then the lookup part : Photons per voxel → PMT response

  New module, PropagatePhotons, to run side by side with DriftElectrons. Do not step any photons, but rather use a custom scintillation process to produce inputs with which to sample the PMT response library

- Computational power to build the library

  Certainly not locally on flxi09. Can somebody give me some pointers?

Summary

- Steady progress on Optical MC
- With the PMTSensitiveDetector and PMTHits in place, framework for simulation and detection of optical photons is now complete.
- Realistic PMT geometry defined, and realistic positions on the way
- Work towards building a PMT response library has begun, first step was development of the LightSource event generator which is now in CVS
- Onwards to library builder, library reader and PMT digizitation